Changepoint Detection in Social Networks: an Extension of the Relational Event Model

RESEARCH REPORT

Hsuan Lee (9252568)

Supervisors:

Dr. Mahdi Shafiee Kamalabad
&
Dr. Javier Garcia Bernardo

Programme: MSBBSS

Department of Methodology & Statistics

Utrecht University
the Netherlands

Date: 08.05.2023
Candidate Journal: Social Networks
FETC Case Number: 22-1870; 22-1871

Abstract

Detecting changepoints is crucial in the social network field to identify the time points when stimulus occur in the network, providing insight into the dynamic evolution of the network. This paper presents an approach that facilitates the detection of changepoints in social networks by combining changepoint detection algorithms with the Moving Window-Relational Event Model (MW-REM). The MW-REM is a social network model that predicts future events and accounts for factors that influence social interactions. To capture the dynamics of the influence levels of these factors over time, the MW-REM uses the Moving Window (MW) approach that models each fitted window to capture fluctuations in the effects' influence level. The approach suggested in this paper incorporates changepoint detection algorithms to detect sudden shifts in the influence levels of these factors using the parameters given by the MW. We conducted an evaluation of three changepoint detection algorithms - Binary Segmentation, Pruned Exact Linear Time, and Bayesian Online Changepoint Detection using synthetic data. To test the practical feasibility of our proposed method, we then applied these algorithms to analyze real-life Apollo-13 voice loop data. The approach can facilitate a better understanding of social dynamics, help identify potential challenges, and inform future strategies based on data.

Keywords:
Social network, Relational Event Model, Moving Window approach, Changepoints detection, Binary Segmentation, Pruned Exact Linear Time, Bayesian Online Changepoint Detection Method

1. Introduction

Changepoints are sudden shifts within the time series data which reflect the transitions occurring across conditions[30][3]. The detection of changepoints is essential in many domains. In the social network area, due to the rich dynamics of social networks, changepoint detection allows us to spot the time points when certain stimulus occurred in the network, such as traffic accidents, organizational interventions, etc., assisting us to observe the dynamic evolution of the network more clearly. However, research and applications of changepoint detection are relatively lacking in the social network field.

The aim of this study is to suggest an approach that facilitates detection of changepoints in social networks. To this end, we introduce changepoint detection methods into the structure of the relational event model (REM). The REM is a social network model that can predict the time and individuals involved in the future events in a social network. It is fitted in the relational event history (REH) data (Table 1), a social network data that at minimum contains information about the time, senders and receivers of events occurring in the social network. Generally, REM brings factors (e.g., gender, age, interaction inertia, etc.) that affect social interactions into the model through parameterization. Such factors are termed "effects"[7], and the parameters of the REM represent the influence level of each effect in the social network.

Time (hh, mm, ss)	sender	receiver	message
13:14:05	Patel	Chen	The weather conditions look good.
13:14:09	Chen	Patel	Yes, it should be a smooth flight.
13:14:12	Nguyen	Chen	I've done my safety checks.
13:14:14	Chen	Nguyen	Great job!

Table 1.. An Example of Relational Event History Data

However, the influence level of effects in REM is assumed to hold constant over time, which seems unrealistic considering the dynamic nature of real-life interactions. To address this limitation, Mulder and Leenders [24] proposed the Moving Window (MW) approach built upon the REM construct, which is capable of capturing the dynamics of the influence levels across time. The main concept of MW is to delineate specific duration of time (i.e. a window) that partially overlaps with the subsequent window and slides over the entire event history. Thereafter, by modeling each fitted window, we can model the fluctuations of the effects' influence level. The difficulty, however, is that the influence level typically varies between consecutive windows, leaving the visual identification of changepoints a challenge.

In this study, we introduce changepoint detection algorithms (CPDs) into the combined MW-REM framework. By feeding these algorithms (influence level) parameters along the windows, we enable researchers to discover changepoints across the REH. This can be employed on any social network scenario, such as: communication in the surgical room, interaction between teacher and students, as well as cooperation and competition between companies. The detection of changepoints can thus facilitate better understanding of social dynamics, help identify potential challenges as they appear and help inform future strategies based on data. Our study focuses specifically on the performance of three CPDs in the MW-REM structure: Binary Segmentation (BS), Pruned Exact Linear Time (PELT), and Bayesian Online Changepoint Detection (BOCPD). These methods were identified as top performers in a comparison conducted by van den Burg and Williams [11] in a general application. To evaluate their feasibility and performance in the context of social network scenarios within MW-REM, we utilize synthetic data to calculate metrics such as Confusion Matrix, Mean Squared Error (MSE), and Mean Signed Difference (MSD). We also apply these methods to real-life data to test their external validity. The blueprint of our study is depicted in Figure 1.

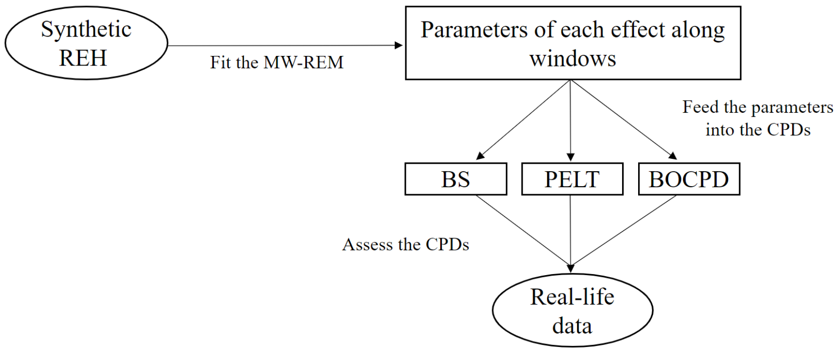

Figure 1.. Flowchart Depicting the Research Design of the Present Study

2. Methodology

2.1. *Moving Window-Relational Event Model (MW-REM)*

In this study, we utilized the Moving Window-Relational Event Model (MW-REM), a hybrid approach developed by Mulder and Leenders[24]. The MW-REM builds upon the Relational Event Model (REM) proposed by Butts[7] and overcomes its limitations by identifying changes in the effect parameters over time.

The REM is a powerful tool for modeling network data that contains sender, receiver, and time information, known as Relational Event History Data (REH). By analyzing the impact of various effects on the social network, the REM enables us to predict the next event in the network. This approach parameterizes both exogenous effects, which are actor characteristics that do not depend on past interactions in the network, such as age or gender, and endogenous effects, which depend on past interactions in the network, such as transitivity or inertia.

The REM considers every possible sender-receiver combination (s, r) in the network as a potential occurrence at time t, and the collection of these pairs at time t forms the risk set, denoted as $R(t)$. The risk set's size for each event is usually $N \times (N-1)$, where N refers to the number of actors in the social network, as each actor can only be either a sender or a receiver, but not both.

The REM models the event rate (λ) to predict which sender-receiver pair (s, r) will be involved in the next event and when it will occur. Each (s, r) pair is equipped with its own event rate for each event and is assumed to remain constant between the time of the present event and the time of the following event. The (s, r) pair with a higher event rate in the risk set R at time t is more likely to occur in the next event. The probability of the (s, r) pair taking place in the next event follows a multinomial distribution, which is given by:

$$P\left((s,r)|t\right) = \frac{\lambda(s,r,t)}{\sum_{R(t)} \lambda(s,r,t)} \tag{1}$$

, where $\lambda(s, r, t)$ represents the event rate of a pair (s, r) and $R(t)$ denotes the risk set for time t.

On the other hand, the duration between two events follows an exponential distribution, which is given by:

$$\Delta t \sim Exponential\left(\sum_{R(t)} \lambda(s,r,t)\right) \tag{2}$$

, where Δt denotes the duration between two events. The higher the total event rate of the risk set, the shorter the Δt.

The event rate is typically considered as a log-linear function of the outcome in REM with specific effects, given by:

$$\log \lambda(s,r,t) = \sum_{p} \beta_p x_p(s,r,t) \tag{3}$$

, where β_p represents the parameter of effects, which expresses the influence level of one effect on the entire social network, and $x_p(s,r,t)$ denotes the statistics, which can be either an exogenous or endogenous effects.

The REM can use event rate modeling to forecast the timing and participants of the next event. However, this method assumes a constant level of influence (i.e., β_p) for all effects throughout the event history, which is unrealistic in dynamic social interactions.

To address the limitations of REM, Mulder and Leenders proposed the Moving Window approach (MW) under the REM framework[24]. In this study, we refer to this approach as MW-REM. The MW-REM involves setting up a fixed-size window, i.e., a fixed length of time, that slides over the entire Relational Event History (REH) data. Each window shares a fixed-size overlap with the previous window, and the REM is fitted to each window (see Figure 2). This approach allows us to reveal the influence level of each effect on the social network over time through the parameters (i.e., β_p) given in each window. Additionally, it enables us to investigate social network dynamics over time, a feature not available in the original REM.

To better understand social network dynamics over time and detect significant changes in the network structure, we employ changepoint detection methods in the MW-REM framework.

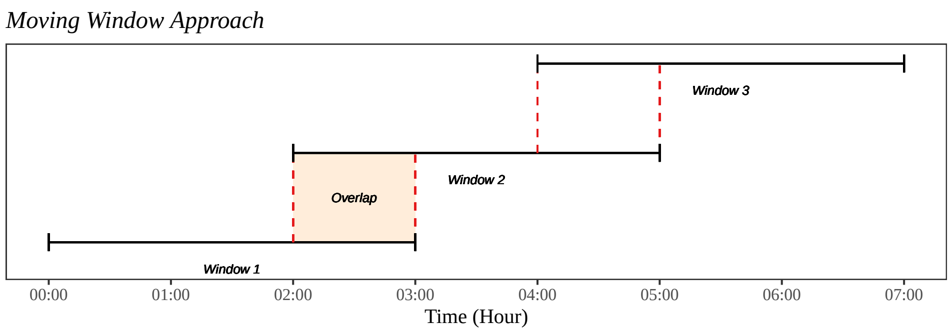

Figure 2.. Example of Moving Window Approach: a window length of 1 hour with an overlap of $\frac{1}{3}$.

2.2. *Changepoint detection*

In social network analysis, changepoints refer to significant changes in the underlying structure of a network over time. Detecting changepoints is crucial for identifying trends, patterns, and events that influence the evolution of social networks and for informing future strategies based on data. Despite its importance, there have been limited studies on inferring changepoints in the REM framework of the social network area. Shafiee Kamalabad suggested using Bayes Factor to infer changepoints on REH of a social network, which uses the support of two hypotheses from the data to prove the existence of changepoints [29]. However, apart from this study, there have been few studies on changepoint detection under the REM framework.

In this study, we propose a novel approach for detecting changepoints in REH, building on the foundation of MW-REM. As MW-REM divides the event history into partially overlapping sub-portions to fit the REM, it provides us with effect parameters (β_p) for each window over time, indicating the influence level of each effect during that period, as shown in Equation 3. By fitting MW-REM to the event history, we obtain as many REMs as the fitting windows, and therefore the same number of effect parameters (β_p) as the number of fitting windows.

Our proposed approach utilizes three CPDs for detecting changepoints in social networks: Binary Segmentation (BS), Pruned Exact Linear Time (PELT), and Bayesian Online Changepoint Detection (BOCPD). BS is an offline method that divides the data into smaller segments and iteratively tests for a changepoint in each segment. It is easy to implement and provides good results in practice. PELT is an online method that uses dynamic programming to search for the most likely changepoints in a computationally efficient way. It has good theoretical guarantees and can detect changepoints in real-time. BOCPD is another online method that uses a Bayesian approach to estimate the probability of a changepoint at each time step, making it effective in detecting sudden changes in the data.

According to van den Burg and Williams11, these three CPDs were identified as top performers in a comparison in a general application. The comparison was based on criteria such as $F1$ scores and Segmentation covering metric. We chose these CPDs for their respective advantages and suitability for detecting changepoints in social networks, which can have abrupt changes over time.

To detect changepoints in social networks, we follow a five-step approach. First, we identify the key effects that drive social interactions in the specific network, such as gender and inertia, etc. Next, based on our knowledge of the network, we select an appropriate window length and overlap for analyzing the event history of the network, for instance, a window length of 2000 seconds with a $\frac{2}{3}$ overlap. In the third step, we fit the MW-REM to the event history then extract the effect parameters (β_p) for each fitting window of each effect. Finally, we apply changepoint detection algorithm to the effect parameters of each effect to identify potential changepoints of the certain effects throughout the social network. The algorithm for this process is summarized in Algorithm 1.

Algorithm 1 MW-REM CPs Detection

1: **Input:** A REH data
2: Select the effects that drive social interactions
3: Select an appropriate window length and overlap (e.g., 2000 seconds with $\frac{2}{3}$ overlap)
4: Fit the MW-REM model to the given REH
5: Extract the parameter (β_p) of each effect from each window's REM: $\log \lambda(s, r, t) = \sum_p \beta_p x_p(s, r, t)$.
6: Apply changepoint detection methods to each effect's parameters to identify potential changepoints
7: **Output:** Potential changepoints for each effect in the social network

The fundamental mechanisms of the three CPDs are presented in the following sections.

2.2.1. Binary Segmentation (BS)

BS is the most commonly used CPD in many research areas. BS recognizes the changepoints by computing the time point in the data that minimizes the given cost function, then divides the data into two segments by it(i.e., the location of the changepoint), after which the same procedures are performed in each separated sub-segment until no changepoint is left in each segment [20].

In changepoint detection, the cost function is a quality measure that splits the data into segments. CPD is designed to segment the data at the time point where the cost function is minimized. The most common cost function for multiple changepoint detection is [20]:

$$\sum_{i=1}^{m+1} \left[C(y_{(\tau_{i-1}+1):\tau_i}) \right] + \beta f(m) \tag{4}$$

, where i denotes the order of a time point in a segment, m indicates the number of changepoints, τ_i implies the location of a possible changepoint (i.e., time point i). And the m changepoints will divide the data into $m+1$ segments, with the ith segment contains $y_{(\tau_{i-1}+1):\tau_i}$. C represents the cost function of a segment, $\beta f(m)$ serves as a penalty to prevent overfitting.

For BS, the recognized changepoints of the data are required to meet the criteria:

$$C(y_{1:\tau}) + C(y_{(\tau+1):n}) + \beta < C(y_{1:n}) \tag{5}$$

BS continues to search for possible changepoints until there is no τ that meets the criteria, then BS stops.

2.2.2. Pruned Exact Linear Time (PELT)

The PELT is a method based on the Segment Neighborhood changepoint detection method (SN)[5]. SN employs dynamic programming to scan the entire segmented area. It first establishes the maximum number of changepoints, next evaluates the cost function of all possible segments. Eventually, the number of changepoints of the data is between 0 and the previously set maximum number. Further, the SN is capable of including any form of penalty(i.e., $\beta f(m)$), yet with the drawback of high computational cost.

The PELT solves the high computational cost of SN whilst maintaining the accuracy of identifying changepoints. It improves computational efficiency by removing the value of τ from each iteration that is unlikely to be the minimum value, simultaneously searching for the value of τ that minimizes the cost function below [20]:

$$\sum_{i=1}^{m+1} \left[C(y_{(\tau_{i-1}+1):\tau_i}) + \beta \right] \tag{6}$$

, which is equivalent to (4), where $f(m) = m$. The PELT considers the cost function of all potential segments, which ranges between 0 and the previously set maximum number of changepoints, and stops when no more changepoints are detected.

2.2.3. Bayesian Online Changepoint Detection (BOCPD)

Unlike BS and PELT, which rely on cost functions to identify changepoints, BOCPD infers changepoints based on Bayesian approach, which define changepoints in terms of posterior probabilities (i.e., run length probabilities) at time points.

In BOCPD, run length is an essential concept, it represents the length of time points elapsed since the last identified changepoint, which can be understood as akin to the segments in PELT and BS. Whenever BOCPD recognizes a changepoint, the run length drops to 0 and recalculates the length. To determine the changepoints, BOCPD is required to calculate the run length probabilities (i.e., posterior probabilities) for each time point. Given that each time point contains probabilities of increase and decrease in run length, the run length probabilities thus includes both growth probabilities and changepoint probabilities. According to Adams and Mackay[1], to save computational costs, it is suggested to set a cut-off point for the run length probability, typically 10^{-4}, where if the run length probability reaches such cut-off point, the time point is determined as a changepoint.

Overall, the BOCPD starts by building the predictive distribution from the potential locations of changepoints, which reveals any prior knowledge regarding the data generation process. Then, based on the given predictive distribution, the BOCPD computes the run length probability at a time point, with new data coming in, the predictive distribution is continuously updated, the BOCPD iteratively runs the same procedure until no new data appear.

2.3. *Generation of synthetic REH data*

To evaluate the effectiveness of our proposed changepoint detection approach for REH data, we generated synthetic datasets with various changepoint settings. For each setting, we created 15 datasets that were simulated based on the MW-REM characteristics. These synthetic datasets were constructed using a social network consisting of 30 actors and 10,000 events.

To construct these datasets, we randomly selected two exogenous effects and two endogenous effects from the most common ones. We then specified the parameter values (β_p) for each effect and its changes over time. The assignment of the effects' parameters in the synthetic data is based on Meijerink-Bosman's [22] with slight adjustments. The exogenous effects include the "Sender effect," which represents the actor's exogenous attributes that impact their event sending rate, and the "Difference effect," which indicates the difference in personal attributes that influence the rate of sending events. The endogenous effects consist of the "Inertia effect," which describes the tendency of actors to repeatedly select the same receiver for their events, and the "Outdegree of the Sender effect," which indicates the inclination of actors to send events if they have previously sent more events.

Using the assigned parameters of the effects, each sender-receiver pair (s, r) in the risk set R obtained the probability of occurrence at every event, as shown in Equation 1. We then built the REH by selecting the (s, r) pair with the highest probability at each event. In this study, we designed five different changepoint settings for the REH data, each of which we simulated 15 datasets:

(1) All of the effects do not have any changepoints.
(2) The inertia effect has 1 changepoint, but the rest do not.
(3) All of the effects have 1 changepoint.
(4) The inertia effect has 2 changepoints, but the rest do not.
(5) All of the effects have 2 changepoints.

For the datasets with the 1 changepoint setting, we set the changepoint for the effect(s) parameter at $t = 38200$ seconds, which corresponds to the window of 56, 57, and 58. For datasets with the 2 changepoints setting, we set the changepoints for the effect(s) parameters at $t = 16750$ seconds and $t = 53900$ seconds, corresponding to the windows of 24, 25, 26, and 79, 80, 81, respectively.

2.4. *The effectiveness and performance of CPDs*

Having generated synthetic REH data with different changepoint settings, we proceeded to evaluate the effectiveness of our proposed changepoint detection approach on these datasets. To this end, we fitted the MW-REM with a window length of 2000 seconds and $\frac{2}{3}$ overlap to each dataset and extracted the effects' parameters. In the following section, we describe how we inspected the performance and compared three changepoint detection algorithms applied to these extracted parameters, utilizing three metrics: the confusion matrix, mean squared error (MSE), and mean signed difference (MSD).

2.4.1. Confusion matrix

After feeding the changepoint detection algorithms with the effects' parameters, we used the confusion matrix to evaluate the performance of each algorithm for each effect. A true positive indicates that the algorithm correctly detected a window containing the changepoint of the effect. Notably, due to the overlapping property of the windows, a changepoint can be contained in three consecutive windows simultaneously. Therefore, if an algorithm detects the changepoint in any of the three windows, it is considered a true positive. However, since the synthetic datasets are produced based on selecting the (s, r) pair with the highest rate probability in the risk set R for every event, the locations of the changepoints may differ slightly from our assignment. As a result, if a changepoint detection algorithm detects a changepoint within the range of three windows of the windows containing the true changepoint, we consider it a true positive.

A false positive indicates that the algorithm falsely detected a window as a changepoint, while a false negative indicates that the algorithm failed to detect a window containing the changepoint of the effect. However, our synthetic datasets are highly imbalanced, with many windows and few changepoints, so we did not take the true negative into account. Table 2 presents the confusion matrix for changepoint detection, showing the actual and predicted changepoints and non-changepoints.

	Actually Changepoint	*Actually Not Changepoint*
Predicted Changepoint	True Positive	False Positive
Predicted Not Changepoint	False Negative	

Table 2.. Confusion Matrix for changepoint detection

Given the information from the confusion matrix, we employ three indicators to assess the effectiveness of our proposed approach and the performance of the three changepoint detection algorithms in detecting changepoints in REH. The first indicator is the number of false positive cases. We separately sum the number of false positives of the effects with no changepoint, one changepoint, and two changepoints for each changepoint detection algorithm. For instance, if the REH has one changepoint in the inertia effect, but not for the other effects, the number of false positives of the inertia effect by a changepoint detection algorithm is summed in the one changepoint group, while the rest of the effects are summed in the no changepoint group. This is done to determine which of the three changepoint algorithms has the highest likelihood of falsely detecting a changepoint when there is none, considering the no changepoint, one changepoint, and two changepoints settings for each effect.

The second indicator is the true positive rate (TPR). Similar to the false positives indicator, we separate the effects into groups based on their number of changepoints. However, since there are no true positives for the effects with no changepoint, we only consider the groups with one and two changepoints for each changepoint detection algorithm. The TPR is calculated as follows:

$$TPR_g = \frac{TP_g}{TP_g + FN_g} \tag{7}$$

, where TP indicates the number of true positive cases, FN indicates the number of false negative cases, and g indicates the changepoint group. Through the TPR, we obtain information about the probability that each changepoint detection algorithm correctly predicts the true positive for the effects with one or two changepoints, respectively, among all positive observations.

The third indicator is the positive predictive value (PPV). Similar to the TPR and false positives indicators, we also group the effects into one changepoint and two changepoint groups based on their number of changepoints. The PPV is calculated as:

$$PPV_g = \frac{TP_g}{TP_g + FP_g} \tag{8}$$

, where TP represents the number of true positive cases, FP represents the number of false positive cases, and g represents the changepoint group. The PPV indicates the probability that a predicted changepoint is indeed a changepoint, providing insight into the precision of the changepoint detection algorithms.

2.4.2. Mean Squared Error (MSE) & Mean Signed Difference (MSD)

To evaluate the performance of each changepoint detection algorithm, we use mean squared error (MSE) and mean signed difference (MSD) as measures to examine the accuracy of the predicted changepoint window and the tendency of the algorithm to detect changepoints early or late.

The MSE measures the average squared distance between the predicted and actual changepoint windows for each changepoint detection algorithm. We compute the MSE only for true positive cases from the confusion matrix of each effect. Specifically, we calculate the MSE as follows [3]:

$$MSE = \frac{\sum_{i=1}^{\#CP} (Predicted(CP) - Actual(CP))^2}{\#CP} \tag{9}$$

,where CP denotes the windows containing a changepoint, and $\#CP$ represents the number of changepoints in the REH. The lower the MSE, the higher the precision of the algorithm. To evaluate the performances of the changepoint detection algorithms across different changepoint scenarios, we group the effects into one-changepoint and two-changepoint groups and report the average MSE of each group for each algorithm using the following formula:

$$Avg.MSE_g = \frac{\sum_{i=1}^{N_g} MSE_i}{N_g} \tag{10}$$

, where $Avg.MSE_g$ represents the average MSE for group g, MSE_i is the MSE value for the ith effect in group g, and N_g is the number of effects in group g.

The MSD, on the other hand, is used to examine the tendency of the changepoint detection algorithm to detect changepoints early or late. We compute the MSD only for true positive cases from the confusion matrix of each effect. Specifically, we calculate the MSD as follows:

$$MSD = \frac{\sum_{i=1}^{\#CP} (Predicted(CP) - Actual(CP))}{\#CP} \tag{11}$$

, here, a negative MSD indicates that the changepoint detection algorithm tends to predict the changepoint window earlier than the true changepoint window, while a positive MSD indicates that the algorithm tends to predict the changepoint window later than the true changepoint window.

To evaluate the changepoint detection algorithms' detection tendencies under different changepoint scenarios, we group the effects into one-changepoint and two-changepoint groups. The average MSD of each group for each algorithm is reported using the following formula:

$$Avg.MSD_g = \frac{\sum_{i=1}^{N_g} MSD_i}{N_g} \tag{12}$$

, where $Avg.MSD_g$ represents the average MSD for group g, MSD_i is the MSD value for the ith effect in group g, and N_g is the number of effects in group g. This allows us to assess each changepoint detection algorithm's tendency to detect changepoints early or late in both univariate and multivariate changepoints scenarios of effects.

2.5. *Manipulation of real-life Apollo 13 voice-loop data*

The study utilized real-life data from the publicly available Apollo 13 voice loop data, which captured the communication between the astronauts and Mission Control during the failed Apollo 13 mission. This mission aimed to land on the Moon, but a routine agitation of one of the oxygen tanks caused an explosion that damaged the wire insulation inside, resulting in the discharge of the contents of both oxygen tanks. This left the astronauts without systems to generate electricity and oxygen, prompting them to contact Mission Control for assistance, and ultimately leading to the cancellation of the mission. The analyzed data covers the period from one hour before the emergency until Apollo 13 was safely back on a trajectory towards Earth, specifically from 54:46:28 to 62:06:53 (hh:mm:ss) of the mission timeline.

In the Apollo 13 voice loop data, a pivotal moment occurred when an astronaut stated, "I believe we've had a problem here," at 55:55:21. This moment marks the start of the emergency, and we have selected it as the location of the changepoint in our study. We hypothesize that the old interaction patterns were disrupted after this point, making it an ideal location to test the effectiveness of our proposed changepoint detection approach on a real social network.

To apply our proposed changepoint detection approach (see Algorithm 1), we selected several effects that have a relationship with the network. These effects include the "Inertia effect," which refers to the tendency for actors to repeatedly interact with each other, the "Outdegree of the sender effect," which refers to the tendency for actors to send events if they have sent more past events, the "Indegree of the receiver effect," which refers to the tendency for actors to receive events if they have received more past events, and the "Total degree of the sender effect," which refers to the tendency for actors to send events if they have sent and received more past events. We also selected three other effects that capture specific patterns of interaction: the "AB-BA pshift," which refers to the tendency for immediate reciprocation, where the next sender is the current receiver and the next receiver is the current sender, the "AB-XA pshift," which refers to a tendency for turn usurping, where the next sender is not in the current event and the next receiver is the current sender, and the "AB-BY pshift," which refers to a tendency for turn receiving, where the next sender is the current receiver and the next receiver is not in the current event.

Considering the findings of Meijerink-Bosman's research [23], we decide to fit the MW-REM with a 1000-second window length and $\frac{2}{3}$ overlap, as it provided a good insight into the dynamic of the social network. We then apply this window setting to the Apollo 13 voice loop data, extract the parameters of each effect along the windows, and feed them to the three changepoint detection algorithms employed in our study.

In an ideal situation, by feeding the parameters of the effects to the changepoint detection algorithms, they could successfully detect the presence of changepoints for each effect in the MW-REM, around or precisely at the windows that contain the message reporting the issue.

3. Results

3.1. *Synthetic REH datasets analysis*

In this section, we present the results of the changepoint detection analysis conducted on the synthetic REH data. As previously described in the methodology section, we summarize the performance of three changepoint detection algorithms applied to the effects' parameters for zero, one, and two changepoints, respectively. We evaluated the algorithms' performance using (1) the number of false positives, (2) true positive rate, (3) positive predictive value, (4) average MSE, and (5) average MSD.

3.1.1. *The number of false positives*

As mentioned in the methodology section, a detected changepoint window is considered a true positive if it is within three windows of the true changepoint window. A detected changepoint window is considered a false positive if it is not within the three-window range of the true changepoint window.

The synthetic data reveals that, for effects without a changepoint, BOCPD produces a total of 27 false positive cases, while BS and PELT produce 49 and 98 false positives, respectively. For effects with one changepoint, BOCPD generates 5 false positives, while BS and PELT produce 13 and 19 false positives, respectively. For effects with two changepoints, BOCPD results in 2 false positives, while BS and PELT produce 13 and 19 false positives, respectively. Overall, as Figure 3 shows, BOCPD performs better than the other algorithms in correctly identifying the window that is not a changepoint window, while PELT is the most prone to incorrectly identifying non-changepoint windows as changepoint windows.

3.1.2. *True positive rate (TPR)*

Regarding TPR, Figure 3 shows that all three changepoint detection algorithms have similar TPR values when there is a univariate changepoint in the effect, with BOCPD having 0.52, PELT having 0.49, and BS having 0.53. However, in the case of multivariate changepoints, the TPR values significantly improve, with BOCPD having 0.75, PELT having 0.79, and BS having 0.73. It is noteworthy that the improvement in TPR values for all three algorithms is substantial when moving from the univariate to multivariate changepoint situation. This indicates that if an effect has only one changepoint throughout the entire REH, the probability of correctly detecting the window within a range of three windows from the true changepoint window using any of the three algorithms is only around 50%. On the other hand, if an effect has two changepoints throughout the entire REH, the probability of correctly detecting the window within a range of three windows from the true changepoint window using any of the three algorithms is above 70%, with PELT almost reaching 80% in the case of multivariate changepoints. However, PELT's high TPR may be a trade-off with its high false positive cases, as it detects more changepoints, which increases the likelihood of covering more true positives in its prediction.

3.1.3. *Positive predictive value (PPV)*

Moving on to PPV, it measures the proportion of detected changepoints that are true positive changepoints (i.e., within a range of three windows from the true changepoint) out of all detected changepoints. In other words, it evaluates the precision of the changepoint detection algorithms. For effects with one changepoint, BOCPD has the highest PPV, followed by BS and PELT, with BOCPD's positive predictive value being 0.89, while BS and PELT have 0.75 and 0.66, respectively. For effects with two changepoints, BOCPD still has the highest positive predictive value of 0.98, followed by BS with 0.95 and PELT with 0.86.

Overall, as shown in Figure 3, there is an improvement in PPV for all three changepoint detection algorithms when moving from the univariate to the multivariate changepoint situation, with all three algorithms having a PPV above 0.85 in the multivariate changepoint situation. This indicates that the probability of a predicted changepoint being within three windows range of the true changepoint is higher than 85% in the multivariate changepoint situation. Furthermore, BOCPD consistently performs at a high level in terms of PPV, whether in the univariate or multivariate changepoint situation, indicating that the changepoints identified by BOCPD have a high probability of being within three windows range of the true changepoints. Conversely, PELT exhibits the worst performance in terms of PPV, which is influenced by its high false positive rate.

3.1.4. *Average mean squared error (Average MSE)*

The average MSE measures the average squared distance between the predicted and actual changepoint windows for different changepoint situations. A lower MSE value indicates greater

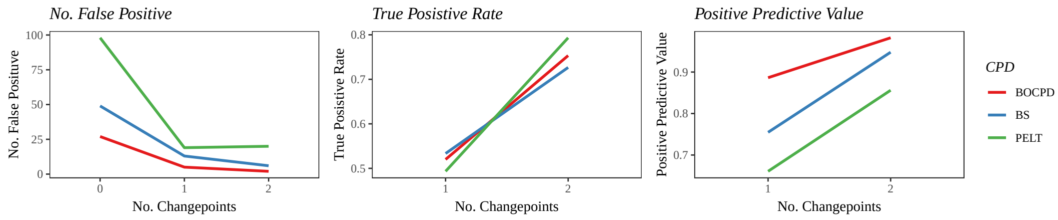

Figure 3.. The plot shows the number of false positives, true positive rate, and positive predictive value of the three changepoint detection algorithms for effects without changepoint, effects with one changepoint, and effects with two changepoints.

precision. In the case of a single changepoint, BOCPD has an average MSE of 1.59, while BS and PELT have MSE values of 1.18 and 1.49, respectively. When there are two changepoints, BOCPD has an average MSE of 1.12, BS has an MSE of 0.76, and PELT has an MSE of 0.94. Overall, BS performs the best out of the three algorithms in both univariate and multivariate changepoint situations, indicating that it has the lowest differences between its predicted and actual changepoint windows (see Figure 4). All three algorithms also perform better in the two changepoint situation, suggesting that their predictions of changepoint window location are more reliable in the multivariate changepoint context compared to the univariate changepoint context.

3.1.5. Average mean signed difference (Average MSD)

The average MSD provides information about how well a changepoint detection algorithm predicts the location of the changepoint window relative to the true changepoint, for different changepoint situations. A negative value indicates that the algorithm is more likely to predict the window before the true changepoint, while a positive value indicates that it is more likely to predict the window after the true changepoint.

In the single changepoint scenario, the BOCPD has an average MSD of 0.05, the BS has -0.03, and PELT has 0.03. In the two changepoint scenario, the BOCPD has an average MSD of 0.12, the BS has -0.1, and PELT has -0.09. From Figure 4, it's clear that the BOCPD tends to detect the changepoint window after the true changepoint, whether in the univariate or multivariate changepoint context. In contrast, the BS tends to detect the window before the true changepoint, and PELT has mixed performance. Interestingly, the reason why BS and PELT tend to detect the changepoint window before the true changepoint is that they are bottom-up methods that merge small intervals based on a cost function balancing data likelihood and model complexity [20]. This approach favors fewer, larger changepoints and may lead to detecting changepoints slightly before the true location. Additionally, it's possible that the true changepoint window is located in a region of the time series where the effect parameters are already changing but have not yet fully transitioned to the new regime.

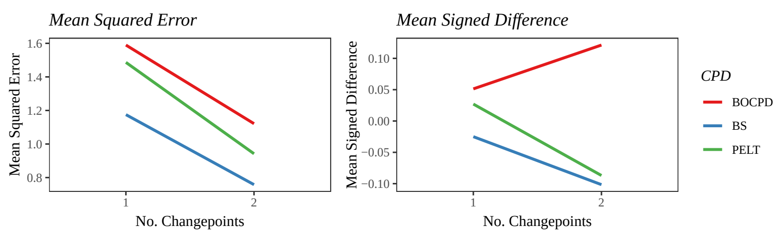

Figure 4.. The plot shows the average mean squared error, and average mean signed difference of the three changepoint detection algorithms for effects with one changepoint, and effects with two changepoints.

3.1.6. *Conclusion*

The synthetic data analysis confirms the effectiveness of our proposed changepoint detection approach for the MW-REM structure. By feeding the changepoint detection algorithms with the effect parameters given by windows, we can identify the points at which each effect changes in the social network.

Regarding the performance of the three changepoint detection algorithms on our method, we can conclude that the BOCPD performs well in terms of PPV, indicating that the changepoints identified by the BOCPD are highly likely to be the true changepoints or within a three-window range of the true changepoint. The PELT has good performance on TPR, indicating that the true changepoints are likely to be covered by all the predicted changepoints from PELT. However, this comes at a cost of high false positives. The BS performs well in terms of MSE, indicating that the location of the predicted changepoint window is close to the true changepoint window.

3.2. *Real-life Apollo 13 data analysis*

The emergency report of Apollo 13 occurred at t = 55:55:21 (hh:mm:ss). The MW-REM was applied with a 1000-second window length and $\frac{2}{3}$ overlap, resulting in a total of 76 windows. The actual changepoint window for our study corresponds to the window from 9 to 11, covering the time slot from 55:38:53 to 56:06:40. Figure 5 depicts the fluctuations in each effect's parameters over the REH in a single figure.

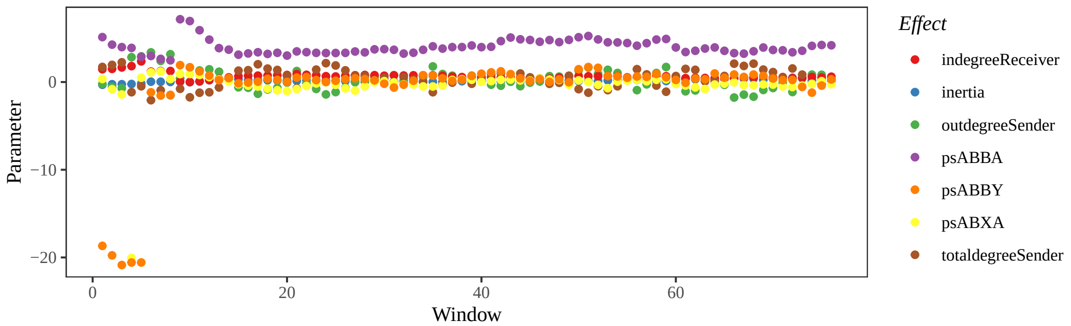

Figure 5.. The figure shows the wave of effect parameters crossing the 76 windows of each of the seven effects we included in the MW-REM.

We inputted the effect parameters obtained from the MW-REM windows into three different changepoint detection algorithms: BOCPD, BS, and PELT. The resulting changepoint detections are displayed in Figure 6. It is worth noting that the emergency happened between windows 9 and 11, so we anticipated that the algorithms would detect the emergency during or around that timeframe.

For the indegree of the receiver effect, BOCPD detected changepoints at windows 9, 15, and 37. BS detected them at windows 6, 8, 12, 15, and 36. PELT detected them at windows 6, 8, 14, 32, and 35. BOCPD successfully captured the changepoint window at 9 and also detected window 15, which occurred four windows later than the actual changepoint window, BS and PELT did not capture any exact window that contained the emergency. BS detected a window that occurred one window before it, and a window that occurred four windows later than it. PELT detected a window that occurred one window ahead of it, and another that was three windows later than it. As we mentioned in the previous section, BS and PELT tend to detect the changepoint window before the true changepoint due to their cost function [20], which favors fewer, larger changepoints and may lead to detecting changepoints slightly before the true location. Additionally, our study is a retrieval study that analyzed the complete data, not incomplete and updated data over time, which may have caused the algorithms to detect the changepoint window slightly earlier.

On the other hand, the fact that the algorithms detected the changepoint slightly after the window that contained the emergency can also be justified. The MW-REM focuses on the dynamics of communication between the actors, in our case, the astronauts and Mission Control. It's possible that their communications were not as frequent immediately after the emergency happened, but instead increased after some time had passed.

From the indegree of the receiver effect in Figure 6, we can also see a significant decrease in the parameter at the window the emergency happened, followed by an increase until window 15 when it stabilized. This indicates that after the issue was reported, the phenomenon of the actors to keep receiving events if they have received more past events decreased, then had a slight rebound. This is consistent with our expectations, as when the issue was reported, the old receivers changed, and the astronauts had more interactions with Mission Control but not between themselves. The staff at Mission Control became the new receivers, and this explains why the parameters rebounded afterward.

For the inertia effect, BOCPD detected changepoints at windows 6, 12, 26, 46, and 60. BS detected changepoints at windows 5, 59, and 73, while PELT detected changepoints at windows 5, 11, 26, 45, 59, and 73. BOCPD detected the changepoint one window after the window containing the emergency report, which is very close to the window where the emergency occurred. However, BS did not detect any relevant windows near the changepoint window. On the other hand, PELT recognized the changepoint at window 11, which is exactly the window where the emergency happened.

The inertia effect in Figure 6 also shows an increase in the inertia parameter after the emergency, which was successfully captured by BOCPD and PELT. The rise in the inertia effect could be because the emergency caused certain astronauts and staff at Mission Control to exchange more reports and responses to tackle the issue during that time period, resulting in a higher inertia effect. However, the effect later decreased slightly.

Regarding the outdegree of the sender effect, BOCPD detected changepoints at windows 4 and 9, BS detected them at windows 3, 8, and 13. PELT detected the changepoint at windows 3 and 8. Once again, BOCPD captured the window in which the emergency occurred, while BS and PELT detected the changepoint one window before it actually happened. The panel of the outdegree of the sender effect in Figure 6 reveals that the parameter of the outdegree of the sender effect had a significant reduction at window 9, the window of the emergency report, and then had a slight increase. This reflects that after the emergency occurred, the phenomenon of the actors who sent more past events would send more in the future was broken. To fix the issue, the astronauts and the staff at Mission Control who were not active senders before had to join the communications more. Therefore, the parameter dropped at window 9, and then they became more active actors than before, leading to a slight rise in the parameter of the outdegree of the sender effect after window 9.

For the AB-BA pshift effect, BOCPD detected changepoints at windows 13, 42, and 60, BS detected them at windows 12, 34, 41, 59, and 73, PELT detected them at windows 13, 34, 41, 59, and 73. Regarding the AB-BY pshift effect, BOCPD detected changepoints at windows 6 and 9, BS detected them at windows 3, 5, and 8, and PELT detected them at windows 3, 5, and 8. For the AB-XA pshift effect, BOCPD detected changepoints at windows 5, 14, 37, and 49, BS detected them at windows 4 and 12, and PELT detected them at windows 4, 13, 36, and 48. Except for BOCPD's performance on the AB-BY pshift effect, none of the three algorithms detected the window that contained the emergency report for these three effects. However, they all detected the window very close to the emergency report window, within a range of three windows from when the emergency occurred. For the AB-BY pshift effect, BS and PELT once again detected the changepoint one window ahead of the emergency report window.

From the AB-BA pshift effect panel shown in Figure 6, it can be observed that after the window containing the emergency, the parameter for the AB-BA pshift effect significantly declined. This suggests a reduction in the immediate reciprocation, where the next sender becomes the current receiver and the next receiver becomes the current sender. Similar to the AB-BA pshift effect, after the window containing the emergency, the parameter for the AB-XA pshift effect also dropped, but not as strongly. This indicates a slight reduction in the pattern where the next sender is not in the current event and the next receiver is the current sender. However, in contrast to the AB-BA pshift effect and the AB-XA pshift effect, the AB-BY pshift effect panel shows an increase in the parameter after the window containing the emergency, suggesting an ascending trend of the phenomenon where the current receiver becomes the next sender and the next receiver is not in the current event. Overall, combining the parameter fluctuations around the changepoints of the three effects,it becomes apparent that the sender was not receiving an immediate response from either the event's receiver or other actors as frequently after the emergency occurred. Instead, the receiver was more likely to become the sender of the following event. This suggests that after the emergency, it may be more common for the astronauts to send messages to the staff at Mission Control, who would then transmit or discuss the plan or solutions with other staff at Mission Control.

Regarding the total degree of the sender effect, BOCPD detected the changepoints at window 4 and 14, PELT detected the changepoints at window 3 and 13. In contrast, BS did not detect any changepoint throughout the entire REH of the total degree of the sender effect. Although none of the algorithms detected the exact window involving the emergency, BOCPD and PELT identified nearby windows, specifically three windows after and two windows later than the emergency report window, respectively. As shown in the Figure 6, the total degree of the sender effect reveals a sudden increase in the parameter following the window that contained the emergency report. This suggests that during the Apollo 13 mission, both the astronauts and Mission Control became more active in sending and receiving messages after the emergency. It is possible that the staff who had more prior communication with the astronauts provided more instructions to help address the issues. The observed phenomenon, where actors send more events if they have sent and received more past events, seems to have played a significant role in the communication dynamics after the emergency.

Overall, we validated our proposed approach for using the MW-REM structure to detect changepoints in social networks with real-life data from the Apollo 13 mission. Specifically, we analyzed the voice-loop data for the emergency report that occurred at t = 55:55:21 (hh:mm:ss). Using a 1000-second and $\frac{2}{3}$ overlap MW-REM, we found that windows 9 to 11 out of a total of 76 windows covered the emergency report. We hypothesized that by inputting each effect's parameters into the changepoint detection algorithm, we could identify the window containing the emergency report. Our analysis confirmed this hypothesis, as all three changepoint detection algorithms we used performed well. They identified the changepoints either at or near the window that contained the emergency report for almost all of the effects.

The Apollo 13 analysis also confirmed the characteristic conclusions of each changepoint detection algorithm as observed in the synthetic data analysis. BOCPD had a high PPV and detected only a small number of changepoint windows, which were mostly the emergency report windows. PELT had the best TPR performance among the three algorithms, but detected a larger number of changepoint windows, including those around or at the emergency report windows for all effects. BS had the worst TPR performance and did not detect any changepoint windows around the emergency for the inertia effect and total degree of the sender effect, while BOCPD and PELT did. On the other hand, both BS and PELT identified the window one step before the emergency report window as a changepoint window for the indegree of the receiver effect, the outdegree of the sender effect, and the AB-BY pshift effect. This is consistent with their performance from the synthetic data analysis, where they possess the negative values of the MSD.

4. Discussion

This paper proposes an approach based on the MW-REM structure to detect changepoints in social networks. Additionally, we compare the performance of three widely used changepoint detection algorithms, BOCPD, BS, and PELT, under the MW-REM structure. The proposed

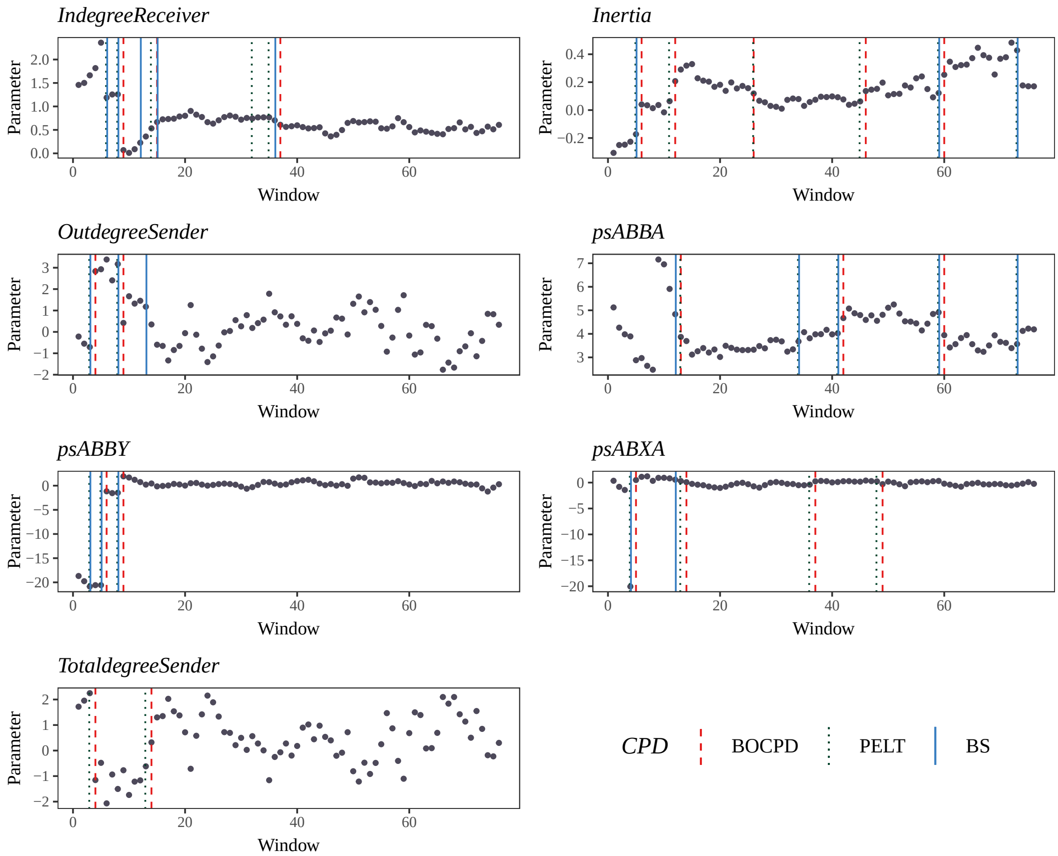

Figure 6.. The plot shows the number of false positives, true positive rate, and positive predictive value of the three changepoint detection algorithms for effects without changepoint, effects with one changepoint, and effects with two changepoints.

approach first identifies the key effects that influence social interactions in the network, and then selects an appropriate window length and overlap to segment the event history of the network. The MW-REM is then fit to the event history, and effect parameters are extracted for each fitting window of each effect. Finally, the extracted effect parameters are inputted into the changepoint detection algorithms to detect the changepoints in the network of each effect. With this approach, we can identify significant changes in the network over time and gain insights into the underlying dynamics of the network.

The validation of our approach using synthetic data and real-life data from the Apollo 13 mission has demonstrated its effectiveness in detecting changepoints in social networks.

Overall, our approach provides a useful tool for detecting changepoints in social networks and can be applied in various fields, such as social science, economics, and epidemiology, to identify critical changes in complex systems. Further research can explore the potential of our approach in identifying changes in different types of networks and comparing the performance of different changepoint detection algorithms under different network structures.

References

[1] R. P. Adams and D. J. C. MacKay. Bayesian Online Changepoint Detection.

[2] R. Alami, O. Maillard, and R. Feraud. Restarted Bayesian Online Change-point Detector achieves Optimal Detection Delay. In *Proceedings of the 37th International Conference on Machine Learning*, pages 211–221. PMLR.

[3] S. Aminikhanghahi and D. J. Cook. A Survey of Methods for Time Series Change Point Detection. 51(2):339–367.

[4] S. Aminikhanghahi and D. J. Cook. A Survey of Methods for Time Series Change Point Detection. 51(2):339–367.

[5] I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. 51(1):39–54.

[6] I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. 51(1):39–54.

[7] C. T. Butts. A Relational Event Framework for Social Action. 38(1):155–200.

[8] C. T. Butts and C. S. Marcum. A Relational Event Approach to Modeling Behavioral Dynamics. In A. Pilny and M. S. Poole, editors, *Group Processes*, pages 51–92. Springer International Publishing.

[9] M. Chapman. A Meta-Analysis of Metrics for Change Point Detection Algorithms.

[10] H. Cho and P. Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. 77(2):475–507.

[11] p. d. u. family=Burg, given=Gerrit J. J. and C. K. I. Williams. An Evaluation of Change Point Detection Algorithms.

[12] p. d. u. family=Burg, given=Gerrit J. J. and C. K. I. Williams. An Evaluation of Change Point Detection Algorithms.

[13] P. Fearnhead and G. Rigaill. Changepoint Detection in the Presence of Outliers. 114(525):169–183.

[14] P. Fryzlewicz. Wild binary segmentation for multiple change-point detection. 42(6).

[15] X. Gu, J. Mulder, and H. Hoijtink. Approximated adjusted fractional Bayes factors: A general method for testing informative hypotheses. 71(2):229–261.

[16] H. Hoijtink, X. Gu, and J. Mulder. Bayesian evaluation of informative hypotheses for multiple populations. 72(2):219–243.

[17] B. Jackson, J. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T. T. Tsai. An algorithm for optimal partitioning of data on an interval. 12(2):105–108.

[18] M. S. Kamalabad, R. Leenders, and J. Mulder. What's the Point of Change? Change Point Detection in Relational Event Models. page 21.

[19] L. Kendrick, K. Musial, and B. Gabrys. Change point detection in social networks—Critical review with experiments. 29:1–13.

[20] R. Killick, P. Fearnhead, and I. A. Eckley. Optimal Detection of Changepoints With a Linear Computational Cost. 107(500):1590–1598.

[21] L. Ma, A. J. Grant, and G. Sofronov. Multiple change point detection and validation in autoregressive time series data. 61(4):1507–1528.

[22] M. Meijerink-Bosman, M. Back, K. Geukes, R. Leenders, and J. Mulder. Discovering trends of social interaction behavior over time: An introduction to relational event modeling: Trends of social interaction.

[23] M. Meijerink-Bosman, R. Leenders, and J. Mulder. Dynamic relational event modeling: Testing, exploring, and applying. 17(8):e0272309.

[24] J. Mulder and R. T. Leenders. Modeling the evolution of interaction behavior in social networks: A dynamic relational event approach for real-time analysis. 119:73–85.

[25] A. Rauhameri, K. Salminen, J. Rantala, T. Salpavaara, J. Verho, V. Surakka, J. Lekkala, A. Vehkaoja, and P. Müller. A comparison of online methods for change point detection in ion-mobility spectrometry data. 14:100151.

[26] M. Shafiee Kamalabad and M. Grzegorczyk. Improving nonhomogeneous dynamic Bayesian networks with sequentially coupled parameters. 72(3):281–305.

[27] M. Shafiee Kamalabad and M. Grzegorczyk. A new Bayesian piecewise linear regression model for dynamic network reconstruction. 22(2):196.

[28] M. Shafiee Kamalabad and M. Grzegorczyk. Non-homogeneous dynamic Bayesian

networks with edge-wise sequentially coupled parameters. 36(4):1198–1207.
[29] M. Shafiee Kamalabad, R. Leenders, and J. Mulder. What is the Point of Change? Change Point Detection in Relational Event Models. 74:166–181.
[30] S. Sharma, D. A. Swayne, and C. Obimbo. Trend analysis and change point techniques: A survey. 1(3):123–130.
[31] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. 167:107299.
[32] G. D. Wambui, G. A. Waititu, and A. Wanjoya. The Power of the Pruned Exact Linear Time(PELT) Test in Multiple Changepoint Detection. 4(6):581.